



(REVIEW ARTICLE)



LIDO: A HUMANISTIC LANGUAGE

Taoufik Ben Hassine *

RIADI Lab, National School of Computer Science, Manouba University Campus, Tunisia.

* Corresponding Author

ORCID Details

Taoufik Ben Hassine: <https://orcid.org/0009-0008-2516-4702>

World Journal of Advanced Research and Reviews, 2026, 31(03), 1878–1881

Publication history: Received on 19 August 2026; revised on 24 September 2026; accepted on 26 September 2026

Article DOI: <https://doi.org/10.30574/wjarr.2026.31.3.2491>

Copyright © 2026 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0

ABSTRACT

LIDO is a domain-specific language (DSL) designed for IoT developers to share their ideas and practical experiences. It consists of an IoT metamodel representing its abstract syntax, and a textual notation representing its concrete syntax. Together, these two syntaxes enable LIDO to convey a horizontal culture shared by the entire IoT community. It also allows developers to leverage their own backgrounds when crafting their own IoT solutions. LIDO was built using an Eclipse Modeling Framework, such as Xtext.

Keywords: DSL, IoT, LIDO, Abstract syntax, Concrete syntax, Xtext.

1 INTRODUCTION

LIDO is an IoT [1] domain specific language (DSL) [2]. LIDO [3] aims to provide a way to express the composition of an IoT solution based on a set of concepts grouped within a metamodel [4]. A concept does not detail a specific solution idea; such a vague idea can be fleshed out in various ways. The value of a metamodel lies in the opportunity it offers to share abstract ideas that can form a collective body of knowledge. The goal is for people to share their reflections with one another—reflections that can serve as a starting point for individuals to conduct their own experiments based on the specific details of their lives. By using the same metamodel, developers can instantiate different models. LIDO merges two approaches that are opposing—if not contradictory: one that favors abstract concepts stripped of detail (since details can hinder sharing and human unity), and another that favors the details that personalize and individualize the experience, thereby making it truly human-centric. LIDO merges these two seemingly contradictory approaches. LIDO consists of an IoT metamodel that constitutes its abstract syntax [5] and a textual notation that represents its concrete syntax [5]. The abstract syntax is obtained through research across dozens of documents to extract the underlying spirit of the IoT. The concrete syntax was generated using a language development environment called Xtext[6].

The remainder of this article is organized into three sections, in addition to the conclusion. The first section focuses on LIDO's abstract syntax. The second section discusses the concrete syntax, and the third deals with the amalgamation—the blending—of the two syntax.

2 THE ABSTRACT SYNTAX OF LIDO

LIDO's abstract syntax consists of an IoT metamodel—a model of models. It serves to describe the components of a model and the relationships connecting them. The metamodel represents the horizontal layer common to all concrete vertical implementations; this horizontal layer constitutes the shared foundation. Composed of concepts that unify the IoT, it acts as a unifying element and a guide for understanding and sharing the essence of the IoT.

A sensor is an example of a concept belonging to the abstract syntax. There are hundreds of sensor manufacturers; these sensors come with various characteristics intended for different uses. All these details are abstracted away by the "sensor" concept itself, making it shareable across all specific sensors and the IoT solutions that employ them.

Another example of a concept is "capability" within the context of IoT devices. Data capture is an example of a specific capability. A "device" is a third example of a concept. A device is linked to capabilities via a "Possesses" or "Has" relationship. A physical object is an object belonging to the physical world. It can be enhanced by a device that endows it with digital capabilities, thereby transforming it into a communicating object capable of receiving and sending messages. These four examples demonstrate that the concepts are very general and can be found in almost any IoT model. When syntax is abstract, it makes it possible to bring together elements of thought, compare them, extract their differences, and integrate them.

Abstract syntax is a kind of syntax for syntax. It is a meta-syntax or a super-syntax and a sort of syntax union. It provides a framework for concrete syntaxes, making their specific characteristics more understandable and worthy of respect. That makes LIDO's abstract syntax adaptable to an IoT model is that abstraction lies at the heart of LIDO. This horizontal layer fosters an IoT culture shared by professionals developing IoT solutions. This is what makes LIDO "humanist," among other things. We believe that humanism is a matter of culture—one that places people at the heart of all development.

3 THE CONCRETE SYNTAX OF LIDO

A concrete syntax is a specific—and therefore vertical—syntax expressed by a model. This vertical model can be derived from the LIDO metamodel or supermodel. The derivation mechanism involves selecting a set of concepts and fleshing them out with the details that specialize in them. The HC-SR501 PIR [7] motion sensor is an affordable and popular passive infrared module used to detect human or animal presence via heat variations. Key technical specifications:

- Supply voltage: 4.5 V to 20 V DC;
- Detection range: Adjustable from 3 to 7 meters;
- Detection angle: Approximately 100° to 110°;
- Output signal: High level at 3.3 V (presence) and low level at 0 V (idle);
- Quiescent current: Less than 50 μ A.

This specific sensor is, after all, an instance of the "sensor" concept or class. It possesses detailed technical characteristics that make it unique and specialized. Taking into account the details and specific features introduced by a developer when building a solution demonstrates respect for their individual culture. This vertical dimension also serves as proof that LIDO is a humanistic language that places human beings at the heart of all development.

We thus move from the general to the specialized; it is a movement of descent and ascent—from the abstract to the concrete, from the horizontal to the vertical, and from the global to the local, and vice versa. Think of the architect and the craftsman: if you want to build a house that everyone appreciates, you need an architect to design a layout that maximizes sunlight, airflow, accessibility, and the view from the outside. That represents the global plan, but the concrete construction details remain the craftsman's secret. This analogy helps IoT newcomers grasp the concepts more easily.

LIDO's concrete syntax is a textual notation. It was created using a Domain-Specific Language (DSL) generator called Xtext. It is against this backdrop that LIDO was designed: to consolidate this convergence and propel it forward. It uses the Eclipse Modelling Framework (EMF) [8] as the core foundation for representing domain-specific language (DSL) models as in-memory abstract syntax trees (ASTs). Thanks to Xtext, the LIDO textual notation—or concrete syntax—features an editor that allows the technical specifications of the devices to be entered into the machine. The editor ensures adherence to concrete syntax rules and offers keyword highlighting and sentence completion.

4 LIDO, THE INTEGRATION AND FUSION OF THE TWO FACETS OF ITS SYNTAX

LIDO was created in the spirit of sharing ideas, concepts, and practical development experiences. IoT aims to digitize the physical world to integrate it into today's digital world. The world is increasingly moving towards a landscape defined by data and information. Distances—across all domains—between countries and communities are shrinking, fostering greater closeness. This shift is driven by the transition from physical to digital realm. LIDO offers two directions for development: a horizontal direction—that of the collective, the community of "being-together"—which fosters sharing and thus brings people together and unites them; and a vertical direction, which allows the developer to express their individuality and specific culture—representing, in Heideggerian terms, "Being" and "beings". Designing an IoT solution with LIDO amounts to selecting the solution's fundamental building blocks. This initial task involves choosing architectural elements, which define the framework or structure of the design. Understanding this structure allows others to grasp the architect's vision. This architecture is a blend of science, art, and tradition—a combination that renders it universal and shareable. The second task involves implementing this architecture using one's own knowledge and expertise. At this stage, a preference for familiar technologies is undeniable. This is the craftsman's approach. Technology underpins construction, and the artisan is increasingly dependent upon it. Although the use of these technologies is highly personal and individual, they are produced and marketed by industries worldwide. LIDO conceptualizes the principles behind them, thereby integrating the global and the local, the universal and the personal; it resolves these two contradictory forces. LIDO supports this dialectic. Although the two facets of LIDO are contradictory, there are two pathways connecting them: a transition from the personal to the universal through abstraction, and a transition from the universal to the individual—or from the global to the local—through concretization.

5 CONCLUSION

LIDO is a DSL. The specific domain in question here is IoT. LIDO comprises a metamodel representing its abstract syntax and a textual notation representing its concrete syntax. The metamodel consists of a set of concepts and relationships between them, derived from a literature review. It aims to help IoT solution developers establish a shared understanding of IoT principles, thereby facilitating the exchange of ideas and practical experiences. It is intended to serve simultaneously as both a universal language and an individual language.

This article explained the merits of its metamodel and the contribution of its textual notation. LIDO was designed after studying a set of metamodels and notations from literature. An example of a model equivalent to LIDO is the metamodel that is part of a European project.

Indeed, LIDO was defined by studying and critiquing—and thus addressing—the shortcomings of IoT-A. IoT-A—the European Lighthouse Integrated Project addressing Internet-of-Things architecture—proposes the creation of an architectural reference model alongside the definition of an initial set of key building blocks. Above all, LIDO provides a context view and a location view, known as "Context-Time-View" and "Place-Anywhere-View". We do not claim that LIDO is superior to European architecture; rather, we maintain that the two overlap, interleave, and complement each other. The abstract and concrete syntaxes that make up LIDO present two facets that appear contradictory in principle but are, in reality, complementary. This article has explained how these two facets combine to establish a dual-level layer, with each level satisfying a human need: the need to be both a universal human being and a unique individual. That is why we titled the article: "LIDO: A Humanistic Language".

REFERENCES

- [1] Shaikh, Aaman & Banerjee, Sudeepta. (2025). "Future of IOT: Emerging Trends", 10.13140/RG.2.2.31886.27209.
- [2] Barisic, Ankica, and Miguel Goulao. "Patterns for Evaluating Usability of Domain-Specific Languages".
- [3] Ben Hassine, Taoufik. (2023). "LIDO: A Universal Language", World Journal of Advanced Research and Reviews, 2023, 18(02), 361–364
- [4] Ben Hassine, Taoufik & Khayati, Oualid & Ben Ghezala, Henda. (2017). "An IoT domain meta-model and an approach to software development of IoT solutions", 32-37. 10.1109/IINTEC.2017.8325909.
- [5] Krahn, Holger & Rumpe, Bernhard & Völkel, Steven. (2007), "Integrated Definition of Abstract and Concrete Syntax for Textual Languages", 10.1007/978-3-540-75209-7_20.
- [6] Zhang, Weixing & Strüder, Daniel & Hebig, Regina. (2025), "Development and Evolution of Xtext-based DSLs on GitHub: An Empirical Investigation", 10.48550/arXiv.2501.19222.

- [7] Naccarelli, Riccardo & Casaccia, Sara & Revel, Gian. (2022), “The Problem of Monitoring Activities of Older People in Multi-Resident Scenarios: An Innovative and Non-Invasive Measurement System Based on Wearables and PIR Sensors”, *Sensors*. 22. 3472. 10.3390/s22093472.
- [8] Richard C. Gronback, “Eclipse Modeling Project: A Domain Specific Language (DSL) Toolkit” (2009), Addison Wesley Professionals, ISBN-10 : 0321534077 ISBN-13 : 978-0321534071